RFC 6029 : A Survey on Research on the Application-Layer Traffic Optimization (ALTO) Problem

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 23 octobre 2010

Date de publication du RFC : Octobre 2010

http://www.bortzmeyer.org/6029.html

Dans le cadre des travaux menés à l'IETF (groupe de travail ALTO <http://www.bortzmeyer.org/alto-wg.html>, RFC 5594[1], etc) et à l'IRTF sur le problème de l'optimisation du transfert de fichiers en pair-à-pair, ce RFC sert de bibliographie : il rassemble des pointeurs, classés et organisés, sur la littérature existante en matière d'optimisation, de recherche du « meilleur pair ».

Le pair-à-pair est un concept difficile à définir et encore plus à mesurer. D'autant plus, même si le RFC ne mentionne pas ce problème, que la répression dont il fait l'objet à l'instigation des ultras de l'appropriation intellectuelle oblige ses utilisateurs à diverses astuces pour dissimuler son utilisation, rendant ainsi les statistiques difficiles. Par exemple, les estimations de la part de trafic qui revient au pair-à-pair varient de 40 à 85 %... Une chose est sûre, cela représente un trafic important, et qu'il est donc logique de chercher à optimiser. Lorsqu'on utilise le pair-à-pair pour des transferts de fichier (avec des systèmes comme BitTorrent), le contenu convoité est répliqué sur de nombreux pairs à travers le monde (section 1 du RFC). Le problème est « Quel pair utiliser ? » Isolé, le pair qui veut récupérer un fichier ne peut utiliser que le hasard, ou bien des mesures qu'il fait lui-même et qui sont souvent imparfaites (par exemple, la mesure de la latence <http://www.bortzmeyer.org/latence.html> avec un ping ne donne qu'une vague idée du débit qu'on obtiendra). La DHT typique ne donne que des optimisations locales, qui ne sont pas forcément bonnes globalement. C'est le problème central de l'optimisation, décrit dans le RFC 5693.

Il y a plusieurs solutions à ce problème, comme des réseaux de machines faisant des mesures applicatives et donnant accès au résultat de ces mesures (Azureus le fait), ou bien comme des fournitures

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5594.txt

d'information aux pairs par le FAI, pour que les destinataires puissent prendre des décisions informées. Ce RFC résume la littérature existante sur ces solutions, en s'appuyant sur le RFC 4981.

La section 2 du RFC examine les résultats de recherche publiés sur ce thème, comme « "*The impact of DHT routing geometry on resilience and proximity*" <http://www.mpi-sws.org/~gummadi/papers/p1101-gummadi.pdf> » de Gummadi, K., Gummadi, R., Gribble, S., Ratnasamy, S., Shenker, S., et I. Stoica, qui étudie la relation entre le réseau "*overlay*" de plusieurs algorithmes de DHT et le réseau physique sous-jacent. Leur conclusion est notamment que les géométries les plus simples (comme l'anneau) sont les meilleures.

Pour les autres résultats de recherche, la section 2 les classe en deux groupes : les méthodes fondées sur des mesures faites uniquement par l'application (comme Meridian <http://www.bortzmeyer.org/meridian.html>) et celles fondées sur une coopération entre les couches (comme P4P).

Les premières sont traitées plus en détail dans la section 2.1. L'idée de base est que les machines mesurent un coût (typiquement, la latence) par rapport à un certain nombre d'amers et qu'on peut ensuite calculer le coût de la communication entre deux machines, par le biais d'un système de coordonnées Internet <http://www.bortzmeyer.org/internet-coordinates.html>. C'est ainsi que fonctionnent IDmaps <http://idmaps.eecs.umich.edu/> ou GNP <http://www.cs.rice.edu/~eugeneng/research/gnp/>. Tous les deux dépendent d'un réseau d'amers stables. Au contraire, Vivaldi <http://pdos.csail.mit.edu/papers/hotnets:vivaldi/> est entièrement distribué et c'est le système utilisé dans le client BitTorrent Azureus.

Les algorithmes cités dans le paragraphe précédent forment une sous-catégorie du groupe « Mesures faites par l'application ». Cette sous-catégorie est celle des systèmes de coordonnées explicites, qui essaient, à coup de mesures, de déterminer la topologie sous-jacente. Dans un réseau aussi complexe que l'Internet, où des inégalités aussi simples que l'inégalité triangulaire (qui dit que le coût pour aller de A à C est inférieur ou égal au coût de A à B plus celui de B à C) ne sont pas forcément vraies, il n'est pas évident qu'une telle approche soit efficace. D'où la seconde sous-catégorie, basée sur une mesure des distances. Moins générales, ces méthodes (dont la plus connue est Meridian <http://www.bortzmeyer.org/meridian.html>) sont peut-être plus réalistes.

Dans tous les cas, toutes les mesures faites par les applications doivent se poser la problème de ce qu'on mesure. Le plus simple est de mesurer la latence (ce que fait, par exemple, la commande ping). Or, des métriques comme la capacité (RFC 5136) ou comme le taux de pertes de paquets (RFC 7680) seraient sans doute plus utiles. Plus difficiles à estimer par des mesures actives, elles conviennent par contre bien aux mesures passives (voir « "*Internet tomography*" <http://www.tsp.ece.mcgill.ca/Networks/projects/pdf/coates_SPMagazine02.pdf> » de Coates, M., Hero, A., Nowak, R., and B. Yu et ou iPlane <http://iplane.cs.washington.edu/>).

Toutes les techniques de la section 2.1, où l'application cherche à se faire une idée de la topologie du réseau souffrent des mêmes problèmes : l'Internet est compliqué (section 3). Par exemple, le chemin le plus court n'est pas forcément le meilleur. Et la topologie ne renseigne pas sur les coûts financiers des liens, par exemple sur le fait que le transit est plus cher que le "*peering*". Enfin, comme toutes les mesures actives, elles ajoutent à la charge du réseau.

Deuxième groupe, dans la section 2.2, les méthodes fondées sur une coopération entre les couches. Ce sont celles où l'application, au lieu de devoir faire les mesures elle-même, peut interroger un service qui a été configuré (typiquement par le FAI) avec des informations sur le réseau, sa capacité, son coût, etc. La plus connue est P4P (section 2.2.1) où des serveurs, les "*iTrackers*", sont interrogés par les applications qui récupèrent ainsi des informations utiles. Des expériences, par exemple celle décrite dans le RFC 5632

indiquent que les gains de temps pour un téléchargement peuvent dépasser un facteur 2. Si le RFC ne parle pas des problèmes politiques, on note que ce mécanisme nécessite que le FAI soit prêt à révéler une partie de la structure de son réseau.

Une autre solution, décrite dans la section 2.2.2, est celle des « oracles » où l'application, au lieu d'avoir accès à des informations sur le réseau, fournit une liste de pairs potentiels et l'oracle lui indique les plus favorables. Là encore, le RFC ne rentre pas dans les questions politiques mais on voit que l'oracle acquière beaucoup d'informations sensibles, notamment sur les pairs avec qui on communique.

Une variante (section 2.2.3) du cas de l'oracle est celui où les deux machines qui veulent communiquer se connaissent déjà mais hésitent sur les adresses IP à utiliser, puisque ce choix influencera le chemin pris (cela ne concerne évidemment que les machines ayant plusieurs adresses). Cette approche, utilisée dans le système IDIPS (voir Saucez, D., Donnet, B., and O. Bonaventure, « "*Implementation and Preliminary Evaluation of an ISP-Driven Informed Path Selection*" <http://inl.info.ucl.ac.be/publications/idips-isp-driven-informed-path-selection> », CoNEXT 2007) pourrait être utilisé pour le pair-à-pair en considérant l'ensemble des pairs possibles comme étant une immense machine virtuelle qu'on cherche à atteindre, un genre d'"*anycast*".

Bien des questions restent ouvertes à l'heure actuelle (section 4). Ainsi, le problème des machines malveillantes qui vont essayer, dans le cas où le système de mesure est distribué, de perturber les mesures ou d'injecter de faux résultats (section 4.2). Dans un réseau de la taille de l'Internet, et compte-tenu du fait que certains ont un intérêt financier à limiter le partage de fichiers, il ne fait guère de doute qu'il y aura de telles machines pirates. Il n'existe pas actuellement de solutions miracles contre elles. Des tests de vraisemblance ont été proposés, pour vérifier si les mesures d'une machine ne s'écartent pas trop de celles des autres. Mais ces tests sont souvent fondés sur des suppositions qui ne sont pas forcément vraies sur l'Internet (comme l'inégalité triangulaire citée plus haut). Dans les cas où les pairs interrogent un oracle, la question de l'authentification de l'oracle est également cruciale. Mais les réseaux pair-à-pair se prêtent mal aux lourdes solutions d'identité comme les PKI.

Une variante de ce dernier problème se pose d'ailleurs au sujet de l'intégrité des données transmises (section 4.3). Ce n'est pas tout d'authentifier le pair ou l'oracle, encore faut-il être sûr que les informations reçues n'ont pas été modifiées en cours de route. Une telle modification permettrait, par exemple, d'affecter un coût très faible à une cible et de faire réaliser ainsi une dDoS contre elle (RFC 4732).

Enfin, parmi les différents problèmes encore largement ouverts, on peut citer les cas où l'optimisation marche trop bien (section 4.6). Poussée à l'extrême, en effet, l'optimisation ferait que les pairs ne s'adressent qu'à des pairs très proches, chez le même FAI et dans la même région. Les grands essaims pair-à-pair se fragmenteraient alors en essaims locaux, n'ayant pas de communication entre eux, ce qui réduirait la redondance qui fait la solidité du pair-à-pair (voir « "*Pushing BitTorrent Locality to the Limit*" <http://hal.inria.fr/inria-00534117/en/> » de Stevens Le Blond, Arnaud Legout et Walid Dabbous).